PROGRESS OUTCOME 4

Robotics challenges

Annotation

Jamal and Roman are able to decompose a robotics problem into logical steps, write an algorithm for the problem, and create a program for a robot. They test whether the program functions as intended when downloaded to the robot and then refine their algorithm and program. In doing so, they show that they are able to use a block-based programming environment to create a program that uses:

- sequence (blocks in the correct order)
- inputs (readings from colour and ultrasonic sensors)
- selection and iteration through the use of repeat-until loops.

They also show that they can:

- refine an algorithm and program to be more precise
- understand connections between programming and algorithms by using an algorithm as code comments to guide and document programming.

Background

The students have been learning to program robots using the LEGO MINDSTORMS® EV3 integrated development environment (IDE). They have practised writing algorithms to decompose problems. They have worked in teams to write, test and debug programs to solve different challenges, using input from various sensors to make decisions about how a robot's motors should move. They are learning the fundamental building blocks of programs (sequence, selection and iteration) so that they can combine these into larger programs to solve a series of problems.

Task

The students are asked to work in pairs to write algorithms for problems that may arise when developing larger programs to solve multi-step robotics challenges.

They are then asked to test their algorithm by translating it to LEGO MINDSTORMS® EV3 code, using the algorithm as code comments and checking to see their program works when downloaded to the robot. The students are asked to refine their algorithm (and program) if bugs are found upon testing.

This task provides an opportunity for students to use the discipline-specific language of computer science.

Student response

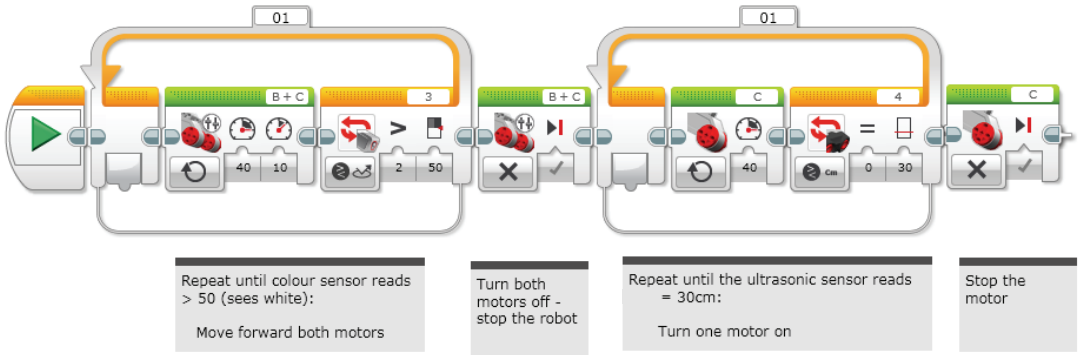
Jamal and Roman formulate their original algorithm for the problem.

With your partner(s) write the algorithm for:

Moving forward until the robot detects white, then turning the robot until an object is detected (e.g. a can) 30 cm or closer and then moving the robot toward that object.

```
Repeat until colour sensor reads > 50:

    Turn both motors on

Turn both motors off

Repeat until the ultrasonic sensor reads = 30cm:

    Turn one motor on

Stop the motor
```

They test their algorithm by translating it into a program that uses input from colour and ultrasonic sensors, selection through use of comparison operators (greater than and equal to), and iteration through a repeat-until loop. Their code comments are taken from their algorithm.



Jamal: *We tested the program, but the robot just spins around and around and never stops.*

Mrs MacKenzie: *What is the logic you are using to stop your repeat loop? Also, have you thought about how the robot will move toward the object once it detects it?*

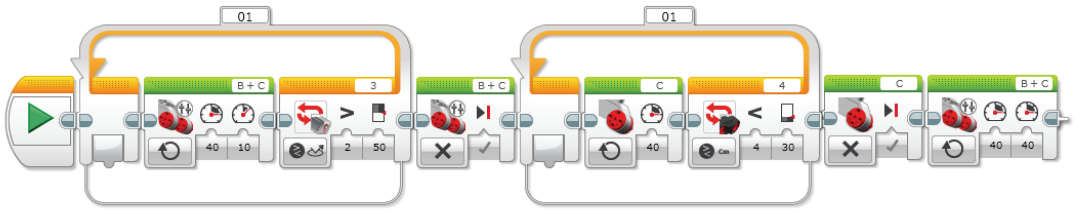
Roman: *We are making our loop continue until it sees an object "equal to" 30 centimetres away. This won't detect anything closer than 30 centimetres. We have to fix our algorithm to look for everything less than 30 centimetres away and add a move-forward step.*

With your partner(s) write the algorithm for:

Moving forward until the robot detects white, then turning the robot until an object is detected (e.g. a can) 30 cm or closer and then moving the robot toward that object.

```
Repeat until until colour sensor reads > 50:

    Turn both motors on

Turn both motors off

Repeat until the ultrasonic sensor reads < 30cm:

    Turn one motor on

Stop the motor

Turn both motors on
```

When they retest their revised algorithm in the program, the robot moves as intended, showing that they have successfully debugged their program using a range of programming constructs.



ISBN: 978-1-77669-203-3

MINISTRY OF EDUCATION
TE TĀHUHU O TE MĀTAURANGA